Tutorial Week 5: SAT

Outline:

1. Encode Propositional Formulas in CNF with **Tsetin Transformation**

2. Model Bit-vector with SAT

3. Problem Solving with SAT: **Palindrome Sum**

Propositional Formulas

Propositional formulas can be defined as:

$$f: \quad \top \mid v \mid \neg f' \mid f_1 \wedge f_2 \quad (\bot \ can\ be\ expressed\ as\ \neg\top)$$

Other operators can be defined as syntax sugars :

$$f_1 \vee f_2 \quad := \neg\,(\neg f_1 \wedge \neg f_2)$$
$$f_1 \Rightarrow f_2 \quad := \neg f_1 \vee f_2$$
$$f_1 \Longleftrightarrow f_2 \quad := (f_1 \Rightarrow f_2) \wedge (f_2 \Rightarrow f_1)$$
$$f_1 \oplus f_2 \quad := \neg(f_1 \Longleftrightarrow f_2)$$

Conjunctive Normal Form (CNF)

- A formula F in CNF is conjunction of clauses:

$$F \coloneqq C_1 \wedge C_2 \ldots C_n$$

where a clause C is a disjunction of literals:

$$C \coloneqq l_1 \vee l_2 \ldots l_m$$

where a literal l is either a variable or its complement:

$$l \coloneqq v \mid \bar{v}$$

From Formula to CNF: Tsetin Transformation

- Sufficient to define transformation for:

1. **Negation**:

$\neg a \quad \rightarrow \quad \bar{a}$

2. **Conjunction**:

$a \wedge b \quad \rightarrow$ a fresh literal $\boldsymbol{c}$ and constraints

$$\left(\overline{a} \vee \overline{b} \vee c\right) \wedge (\bar{c} \vee a) \wedge (\bar{c} \vee b)$$

Modeling Bit-vectors with SAT

Bit-vector (BV) is an array of 0 and 1s.

BV 1110 has the value of 14

We can model BV operations in CNF:

$\mathrm{bv}_1 = bv_2$ $\quad\quad bv_1 + bv_2$

Bit-vector Equality

$bv_1 = bv_2$ iff

$$bv_1[0] \Longleftrightarrow bv_2[0]$$
$$bv_1[1] \Longleftrightarrow bv_2[1]$$
$$\dots$$
$$bv_1[k] \Longleftrightarrow bv_2[k]$$

Bit-vector Addition (no overflow)

$bv_1 + bv_2$ returns a new bit-vector bv_3 where (ripple-carry adder)

$$bv_3[0] = bv_1[0] \oplus bv_2[0]$$
$$bv_3[1] = bv_1[1] \oplus bv_2[1] \oplus carry_1$$
$$bv_3[2] = bv_1[2] \oplus bv_2[2] \oplus carry_2$$

...

$$bv_3[k] = bv_1[k] \oplus bv_2[k] \oplus carry_k$$
$$bv_3[k+1] = carry_{k+1}$$

$carry_{i+1} = AtLeastTwo(carry_i\ , bv_1[i], bv_2[i])$

Bit-vector in Action

Let's prove the following statement with SAT:

$$bv_1 > bv_3 \land bv_2 \geq bv_4$$

$$\Rightarrow$$

$$bv_1 + bv_2 > bv_3 + bv_4$$

Palindrome Sum

Given a natural number n, find two bit-vectors bv_1 and bv_2 such that

1. bv_1 and bv_2 are both palindromes:

   symmetrical, allow padding 0s to the left

2. The value of bv_1 + bv_2 is n

Solving Palindrome Sum with SAT

We already know how to encode bit-vector addition and equality!

We know the bit-vector representation of the input n!

We still need to model the conditions for palindrome:

We need to consider different sizes of paddings for each bit-vector

Finally, we need to call a SAT solver!